--Spesifikasi dan Definiisi konstruktor

konso :: Int->[Int]->[Int]

{-konso e l menambah elemen e ke dalam list l dengan prekondisi l tidak kosong-}

--Realisasi konstruktor

--Spesifikasi dan Definisi Predikat

isOneElmt :: [Int]->Bool

isEmpty :: [Int]->Bool

{-

isEmpty l mengecek apakah list l kosong/tidak

isOneElmt l mengecek apakah list 1 elemen/tidak, prekondisi l tidak kosong

-}

--Realisasi Predikat

isOneElmt l = length l == 1

isEmpty l = null l

--Definisi dan spesfikasi selector

--tail :: [Int]->[Int]

--last :: [Int]->Int

--init :: [Int]->[Int]

--head :: [Int]->Int

{-

tail l mengembalikan l tanpa elemen ke-1

last l megembalikan elemen terakhir l

init l mengembalikan l tanpa elemen terakhir

head l mengembalikan elemen ke-1 l

-}

--Definisi dan Spesifikas fungsi

{--}

--Realisasi

No 1.

hitungBiaya :: Int -> Int -> Int -> Bool -> Int

hitungBiaya k h s a =

let biayaKamar =

if a then

if k==1 then 600000

else if k==2 then 480000

else 320000

else

if k==1 then 750000

else if k==2 then 600000

else 400000

biayaDokter =

if a then 1350000

else 1500000

in

biayaKamar \* h + biayaDokter \* s

No 2.

ekstrakTanda :: [Int]->[Int]

konso :: Int->[Int]->[Int]

konso e l = [e]++l

ekstrakTanda l =

    if null l then []

    else if head l > 0 then konso 1 (ekstrakTanda (tail l))

    else if head l < 0 then konso (-1) (ekstrakTanda (tail l))

    else konso 0 (ekstrakTanda (tail l))

main = do

print(ekstrakTanda [80,1,0,-1,3,-13])

No 3.

konso :: Int -> [Int]-> [Int]

pisahDua :: Int -> [Int] -> ([Int],[Int])

konso e l = [e]++l

pisahDua n l =

    if null l then ([],[])

    else

        let

            (l1,l2) = pisahDua n (tail l)

        in

            if (head l) >= n then (l1,konso (head l) l2)

            else (konso (head l) l1, l2)

No 4.

konversiSuhu :: Float -> (Float->Float) ->Float

CtoK :: Float ->Float

RtoC :: Float -> Float

CtoK t = 273+t

RtoC t = (5/4) \* t

konversiSuhu t f = f(t)

--Aplikasi dalam lambda

konversiSuhu 25 (\x->x+273) –Kelvin

konversiSuhu 16 (\x->x\*4/5)–Reamur

No 5.

deleteIf :: [Int]->(Int->Bool)->[Int]

isLebih5 :: Int -> Bool

isBetween :: Int -> Bool

isEqual :: Int -> Bool

konso :: Int -> [Int] -> [Int]

konso e l = [e]++l

isLebih5 x = x>5

isBetween x = x>=0 && x<= 100

isEqual10 x = x==10

deleteIf l f =

if null l then []

else if f(head l) then deleteIf (tail l) f

else konso (head l) (deleteif (tail l) f)

--Aplikasi Lambda

deleteIf [2,3,5,4,13,3] (\x->x>5)

deleteIf [2,3,5,4,13,3] (\x->x/=10)

No 6.

nilaiEkstrim :: [Int]->(Int,Int)

nilaiEkstrim l =

if isOneElmt l then (head l, head l)

let (max,min) = nilaiEkstrim(tail l)

in

if (head l)>=max then ((head l), max)

else (min, (head l))